CHAPTER 4

ASSESSING RELIABILITY AND AVAILABILITY OF C4ISR FACILITIES

4-1. Purpose of the assessment

As systems become more and more complex, good methods for specifying and analyzing the systems and their sub-systems become more important. Reliability modeling (including prediction, evaluation, and control) is vital for proper design, dependable operation, and effective maintenance of systems. The popularity of designing redundancy into systems poses additional challenges to reliability professionals. For the various kinds of redundant systems, the reliability and availability are extremely sensitive to even small variations in certain parameters; thus, understanding and insight can be gained only by modeling. The purpose of this section is to provide the reader with an understanding of a type of modeling to assist in the decision making process for facility improvement. The Case Study provides an example of how much effort is necessary to baseline your facility and identify potential improvement areas. It is not specifically designed as an instructional document but more of a tutorial. The Case Study makes use of one software tool to quantify availability. In reality, the Case Study is an explanation of a specific type of modeling designed to predict facility performance, not necessarily simulate facility performance.

4-2. Approach

This modeling approach provides facility managers with a cost effective analysis for baselining their facilities and identifying areas of weakness. The results of the model will provide the basis for trade-offs for improving system availability. This chapter first provides the reader with basic concepts of modeling and identifies two approaches: empirical and statistical. Recommendations as to the type of model will be discussed along with a Case Study of an electrical system analysis utilizing a specific deterministic modeling program called GO.

4-3. General modeling concepts

The need to assess the reliability, availability, and maintainability of a system is becoming more important as organizations understand the potential effects of failures and downtime for the systems. Regardless of what product/service is being offered, or who the intended customer may be, it should be a reasonable assumption to state that the degree of product/service success is directly related to the ability of that product/service to meet or exceed customer expectations. Two popular means of assessing the reliability, availability, and maintainability of a system are empirical methods and statistically-based methods. When the statistically-based reliability characteristics (underlying distribution of failures and distribution parameters) of system modules are known, simulation is one way of using those characteristics to analyze system behavior.

a. Empirical prediction methods. For many types of parts, such as bearings and gears, engineers have noted a relationship between reliability and stress. For example, the bearing industry developed an empirical equation in the 1940s that relates fatigue life and bearing loading. The equation was developed using data analysis techniques such as regression analysis. The equations are valid only for the specific type of part for which they were developed and the type of stress (e.g., fatigue failures). The equations are not statistically based and can provide only point estimates of reliability.

b. Statistically-based prediction models. By collecting data from a sample, either from a test or from field operation, a set of statistics can be derived. Based on the sample statistics, conclusions can be drawn regarding the population from which the sample was taken. When times to failure for parts are recorded, for example, Weibull analysis can be used to determine the statistical reliability function for the sample of parts. If the sample was representative of the population, we can infer the reliability characteristics of the population.

c. Simulation. Over the years, simulation has become a trusted tool in system design, development, implementation, and improvement. Simulation has proven to be versatile, as it can be applied from evaluating theoretical concepts to sustaining an examination of minor improvements to an existing, fully operational system.

(1) Webster's dictionary defines simulation as the following: "the imitative representation of the functioning of one system or process by means of another <a computer ~ of an industrial process>." This definition illustrates why simulation has historically proven to be agreeable to the reliability field, specifically the system parameters of reliability, maintainability, availability, system effectiveness, cost, and schedule.

(2) The eight-step process shown in table 4-1 should be adhered to during a simulation study (when applied to a reliability analysis). Validation is to be carried out throughout the eight-step process.

Table 4-1. Steps in performing a simulation.

Problem Definition: define simulation problem and its objectives.
Model Building: description of system's entities and their interaction.
Data Collection: quantify probability distributions for system's entities.
Program Code: select programming language to execute simulation (best to do before model is completed).
Verification: check that code is achieving expected results.
Experimental Design: determine initial conditions, simulation period and number of runs (must be statistically valid).
Implementation: run simulation and test its sensitivity to variations.
Documentation: document simulation study to verify problem definition objectives are reached (document enough for functional model in future).

4-4. Prediction

There are many valid reasons for predicting reliability. One purpose for reliability prediction is to assess the product design progress and to provide a quantitative basis for selection among competing approaches or components. In addition, prediction results can be used to rank design problem areas and assess trade study results. A combination of prediction methods should be used to assess progress in meeting design goals, achieving component or part derating levels, identifying environmental concerns, controlling critical items and determining end-of-life failure mechanisms. Predictions should be an ongoing activity that start with the initial design concept and the selection of parts and materials, and continue through the evaluation of alternate design approaches, redesigns, and corrective actions. Each iteration of prediction should provide a better estimate of product reliability as better information on the product design approach becomes available. Later predictions, during the developmental phase, are used to evaluate life-limiting constraints, as well as identify design problem areas.

4-5. The GO method

The following sections outline vital information for an analyst utilizing the GO software tool. Paragraph 4-5a reviews the background of the GO software and 4-6b identifies a data source that can be used as an input to the tool. Paragraph 4-5c identifies the logic behind an analysis completed within GO, specifically the operators utilized as a representation of the components of a system. Next, paragraph 4-6 presents a case study, outlines the steps required to complete a GO analysis from identifying the one line diagram for the system to performing an analysis or troubleshooting the GO model. Paragraph 4-7 identifies the bibliographical sources utilized within this paper. Paragraph 4-8 defines important terms concerning the operators utilized within the GO software.

a. GO background. The estimation of product reliability requires judgment about its future. Such predictions are based primarily on modeling past experience and data. The GO software has proven to be a successful means of determining the availability and reliability of systems. GO was first introduced as a means of evaluating key reliability metrics of nuclear power facilities, but over the years it has proven to be a valuable tool in evaluating other systems.

(1) The GO software was originally designed to address the need of measuring the availability of nuclear facilities. The GO method, unlike fault tree analysis which focuses on a single system event and uses good/bad elements, is a comprehensive system analysis technique that addresses all system operational modes and is not restricted to two-state elements. GO is not a simulation package but a tool that utilizes the point estimates of component reliabilities to calculate desired system metrics. The GO procedure has been enhanced over the years to incorporate some special modeling considerations, such as system interactions and dependencies, as well as man-machine interactions. GO models are developed in a forward-looking manner following normal process flow or operational sequences. The models determine all system response modes (i.e. successes, failures, prematures, etc.).

(2) GO models consist of arrangements of GO operator symbols and represent the engineering functions of components, subsystems, and systems. The models are generally constructed from engineering drawings by replacing engineering elements (valves, motors, switches, etc.) with one or more GO symbols that represent system functions, logic, and operational sequences. The GO software uses the GO model to quantify system performance. The method evaluates system reliability and availability, identifies fault sets, ranks the relative importance of the constituent elements, and places confidence bounds on the probabilities of occurrence of system events reflecting the effects of data uncertainties.

(3) Key features of the GO method are:

- Models follow the normal process flow;
- Most model elements have one-to-one correspondence with system elements;
- Models accommodate component and system interactions and dependencies;
- Models are compact and easy to validate;
- Outputs represent all system success and failure states;
- Models can be easily altered and updated;
- Fault sets can be generated without altering the basic model;

- System operational aspects can be incorporated; and
- Numerical errors due to pruning are known and can be controlled.

(4) The GO procedure uses a set of standardized operators to describe the logic operation, interaction, and combination of physical equipment and human actions. The logic for properly combining the inputs for each GO operator is defined in a series of algorithms contained in the GO computer codes. These standardized operators are used to model commonly encountered engineering subsystems and components. A system is modeled by selecting the GO operators that characterize the elements of the system (i.e. represent the operational states that can be taken) and interrelating their inputs and outputs. The specific probabilities of component operation are defined separately as inputs to the computer code.

b. Input data sources to GO models. The U.S. Army Special Mission Office's Power Reliability Enhancement Program (PREP) sponsored a study of the reliability, availability, and maintainability characteristics of 234 power generation, power distribution, and heating, ventilation, and air conditioning (HVAC) items. This study will be summarized and published in the forthcoming Institute of Electric and Electronics Engineers (IEEE) Gold Book. The Reliability Analysis Center, a U.S. Department of Defense Information Analysis Center operated by IIT Research Institute, Rome, NY, began the work in October 1991 and delivered the final report in early 1994. (This study resulted in a publication within IEEE Transactions on Industry Applications in March/April 1999 entitled "Operational Maintenance Data for Power Generation Distribution and HVAC Components", and again in Jan/Feb 2001 entitled "Survey of Reliability and Availability of Power Distribution, Power Generation, and HVAC Components for Commercial, Industrial, and Utility Installations," which article will appear in its entirety as an appendix to the Gold Book). Items that were included in the study are gas turbine generators, diesel engine generators, switch-gear assemblies, cables, boilers, piping, valves, and chillers. This program was designed to determine the effects of "new technology" equipment (i.e. equipment installed after 1971) on availability. Information was obtained on a variety of commercial and industrial facility types (including office buildings, hospitals, water treatment facilities, prisons, utilities, manufacturing facilities, schools, universities, and bank computer centers), with varying degrees of maintenance quality. Data collection guidelines and goals were established to ensure that sufficient operational and maintenance data were collected for statistically valid analysis. Two keys to the data collection process were ensuring data completeness and accounting for maintenance policies.

(1) Data was categorized into different levels of data completeness to ensure that the final data collection included a fair data representation for each component, the data completion was quantified by four levels:

(a) Level 1 - Perfect Data: Data needed for a valid, complete reliability study, including a parts list, failure history data with time-to-failure statistics, parts description data, operational periods, and ten continuous years of recorded data. No engineering judgment or data extrapolation is required.

(b) Level 2 - Not Perfect Data: No serious flaws in data, but data collection process demanded additional time to ensure useful information was gathered.

(c) Level 3 - Verbal/Inspection Data: Serious gaps existed in data that required additional documentation and verification prior to its inclusion in the database. Senior maintenance personnel were interviewed to extract the necessary information to fill the data gaps.

(d) Level 4 - Soft Data: Data that relied on the memories of experienced maintenance personnel from the participating facility; it was often extracted from log books containing maintenance personnel entries, filing cabinets with work order forms, and repair records when outside repair support was needed. Engineering judgment was often required to determine numerous performance parameters.

(2) The data collection effort was planned to minimize the effects of maintenance policies and procedures on the calculated availability values by collecting data from a variety of locations with varying maintenance policies. Each facility's maintenance policy and procedure was categorized into one of three levels:

(a) Above average: The facility not only followed a scheduled, preventative maintenance policy that was equivalent or similar to the manufacturer's suggested policy, but also went beyond it, such as using redundant units, specialized equipment tests (thermograph, vibration analysis, oil analysis), complete spare parts kits for equipment, and so on.

(b) Average: Facility used either in-house maintenance crews performing scheduled, preventative maintenance according to the equipment manufacturer's suggested PM schedule or a combination of in-house maintenance crews and outside contractors. In either case, it was verified that they did follow a fairly rigid schedule.

(c) Below average: Facility's actual policy was less than average. It may have instituted a scheduled maintenance policy but not followed it or it may have had no maintenance policy. Symptoms such as leaky valves with rags tied around them, dirty air filters, squeaky bearings, loose belts, and general housekeeping because of unavailable manpower were typical signs that maintenance at a facility was less than desirable.

c. Input data sources to GO models. The IEEE Gold Book is another source of reliability information which has become the standard for reliability calculations over the years. Chapter 3 of the Gold Book summarizes years of survey information collected from a variety of users and manufactures. This data is compiled into a publication sold around the world. The data is presented in a variety of formats for the analyst discretion in utilizing the information. Reliability information as well as failure mode distribution can be found in chapter 3.

d. GO logic. The basic building block of a GO model is the node. Nodes can be either logical or physical, depending on their model function. Physical nodes correspond to actual pieces of equipment and have a reliability associated with them. They have either zero or one input signals (i.e. they may or may not be dependent on prior equipment for successful operation) with exactly one output signal. Logical nodes correspond to interconnections within the system or represent constraints imposed by mission requirements. Logical nodes have multiple inputs and may have multiple outputs to combine the physical nodes of the system. Examples of the operators found in the GO program are outlined in the following subsection.

e. GO operator types. Each of the following subsections presents all pertinent information for an operator type. Each type is presented with its usual name, symbol, the required operator data, the required kind data, the exact logical operation of the type, and comments. Prior to reviewing the description of the operator types that follow, the table of definitions that appears in appendix E should be read and understood. The following symbols are used consistently (other symbols will be defined as they are used):

S_1, S_2, … the identification number of an input (source, stimulus or input) signal

R_1, R_2, …the identification number of an output (result, response or output) signal

K the kind identification number
VS_i, VR_i the value (time) of signal S_i or R_i
$P_1, P_2, \ldots$ probability
∞ infinity or never

(1) When a type has only one input (output) the subscript on S(R) will be deleted.

(2) The operator and kind data are shown in the same order in which they must appear on data entries. We have generally separated the data items by a comma and a blank, but any combination of blanks and/or comma is permitted. Each record must end with a terminator (a dollar sign or slash depending upon the computer used). These terminators are not shown here.

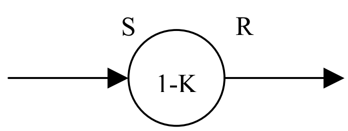
(3) Certain types (6, 7, and 9) have non-symmetric inputs (i.e. inputs are not interchangeable as in type 2 and 10). To differentiate between such inputs on the GO chart symbol we use a full arrowhead for the primary input and a half arrowhead for the secondary one or indicate the primary input by the letter "a" and the secondary one by the letter "b."

(4) The general order of operator data is: type, kind, number of inputs, inputs, number of outputs, outputs. The number of inputs and the number of outputs are omitted when the type definition requires a specific number (i.e. a type 1 always has one input and one output, therefore the two "1's" are not explicitly included in the data list).

(5) Types 2, 10, and 11 do not require kind data. The kind number in the operator data list is set to 0 for types 2 and 10 and is set equal to the value of the extra parameter for a type 11.

f. Type 1: Two state component

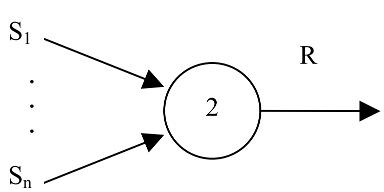
(1) GO symbol:



(2) Operator data: 1, K, S, R
(3) Kind data: K, 1, P_1, P_2
P_1: Component is good
P_2: Component fails
(4) Operation: VR = VS, if the component is good
= never, if the component is not good
(5) Comment: This type models any device which can assume one of the two states. The usual state interpretations are "good" and "bad."

g. Type 2: OR gate

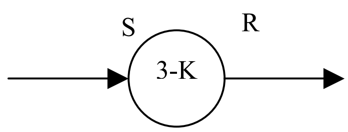
(1) GO symbol:



(2) Operator data: 2, 0, n, $S_1, \ldots, S_n$, R
n: number of inputs, $2 \le n \le 10$
(3) Kind data: none
(4) Operation: $VR = \min \{VS_1, \ldots, VS_n\}$
(5) Comments:
(a) The name "OR gate" is used in the sense that R will occur as soon as S_1 or … or S_n occurs.
(b) Note that the kind number in the operator data is set to zero.

h. Type 3: Triggered generator

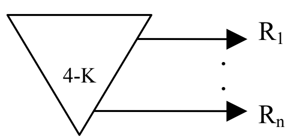
(1) GO symbol:



(2) Operator data: 3, K, S, R
(3) Kind data: K, 3, P_1, P_2, P_3
P_1: generator is good
P_2: generator fails
P_3: generator operates prematurely
(4) Operation: VR = 0, if the actuator prematures
= never, if the generator fails
= VS, if the generator is good
(5) Comment: This type is commonly used to model relay coils, accelerometers, etc.

i. Type 4: Multiple signal generator

(1) GO symbol:



(2) Operator data: 4, K, n, $R_1, \ldots, R_n$
n: number of outputs, $1 \le n \le 10$
(3) Kind data: K, 4, n, m, $V_{11}, \ldots, V_{1n}$, P_1

$$\begin{array}{ccc} \vdots & \vdots & \vdots \\ V_{m1}, \ldots, & V_{mn}, & P_m \end{array}$$

m: number of states for each signal (the total amount of kind data cannot exceed 100 items)
V_{ij}: the value of the i^{th} state of the j^{th} signal
P_i: the probability that the signals are in the i^{th} state

$$\sum P_i = 1.0$$

(4) Operation: $VR_1 = V_{11}, VR_2 = V_{12}, \ldots, VR_n = V_{1n}$ with probability P_1

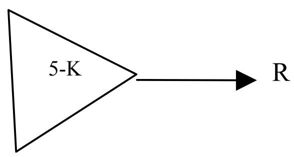
$$\begin{array}{ccc} \vdots & \vdots & \vdots \end{array}$$

$VR_1 = V_{m1}, VR_2 = V_{m2}, \ldots, VR_n = V_{mn}$ with probability P_m

(5) Comment: The type 4 operator generates two or more statistically dependent signals. It is a special case of the type 13 operator.

j. Type 5: Signal generator

(1) GO symbol:



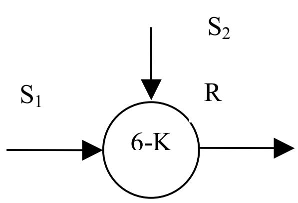
(2) Operator data: 5, K, R
(3) Kind data: K, 5, n, V_1, P_1, …, V_n, P_n
n: number of values for which a signal is to be generated
V_j: i^{th} value
P_i: probability for the i^{th} value

$$\sum_{i}^{n} P_i = 1$$

(4) Operation: VR = V_i with probability P_i, i = 1, …, n
(5) Comment: none

k. Type 6: Normally open contact

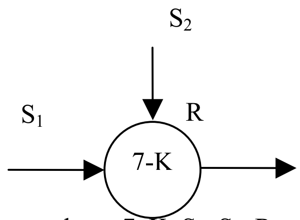
(1) GO symbol:



(2) Operator data: 6, K, S_1, S_2, R
(3) Kind data: K, 6, P_1, P_2, P_3
P_1: contact closes normally
P_2: contact fails to close
P_3: contact closes prematurely
(4) Operation: VR = max {VS_1, VS_2}, if the contact operates normally
= VS_1, if contact closes prematurely
= never, if contact fails
(5) Comment: none

l. Type 7: Normally closed contact

(1) GO symbol:



(2) Operator data: 7, K, S_1, S_2, R

(3) Kind data: K, 7, P_1, P_2, P_3

P_1: contact open normally

P_2: contact fails to open

P_3: contact opens prematurely

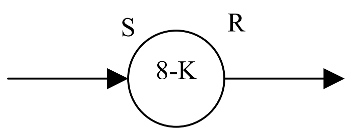
(4) Operation: $VR = VS_1$, if the contact fails, or if $VS_2 > VS_1$ and the contact opens normally

= never, otherwise

(5) Comment: Note the convention that the simultaneous occurrence of S_1 and S_2 produce R at time never.

m. Type 8: Increment generator

(1) GO symbol:



(2) Operator data: 8, K, S, R

(3) Kind data: K, 8, n, D_1, P_1 …, D_n, P_n

N: number of possible increments, $1 \leq n \leq 48$

D_i: value of the i^{th} increment, $-\infty \leq D_i \leq \infty$

P_i: probability that the i^{th} increment occurs

$$\sum_{i}^{n} P_i = 1$$

(4) Operation: with probability P_i, i = 1, n

$VS + D_i$, if $0 \leq VX + D_i < \infty$

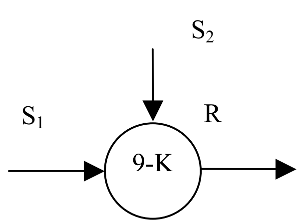
$VR = 0$, if $VS + D_i < 0$

∞, if $VS + D_i > \infty$

(5) Comments: The delay values can be negative as noted in the kind data.
The type 8 operator models component response delays.

n. Type 9: Function operator

(1) GO symbol:



(2) Operator data: 9, K, S_1, S_2, R

(3) Kind data: K, 9, n, X_1, Y_1, …, X_n, Y_n

n: number of X_i, Y_i pairs

X_i, Y_I: ± time values. The set of pairs defines Y_i as a function of X_i (i.e., $Y_i = f(X_i)$). Both X_i and Y_i may lie in the range from –never to +never inclusive. Values of X_i within that range which are not explicitly included in the kind data have an associated Y_i of never.

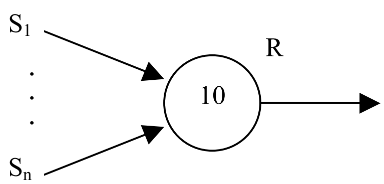
(4) Operation: $VR = \max\{0, \min\{VS_1 + f(VS_2 - VS_1)\}\}$

(5) Comments: This type is "perfect" in the sense that there is always just one output term (with probability 1).

It is used to handle complex timing situations. It is somewhat difficult to get acquainted with but has proved to be of great value in many cases.

o. Type 10: AND gate

(1) GO symbol:



(2) Operator data: 10, 0, n, S_1, …, S_n, R

n: number of inputs, $2 \le n \le 10$

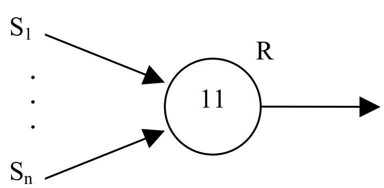
(3) Kind data: none

(4) Operation: $VR = \max\{VS_1, \ldots, VS_n\}$

(5) Comments: The name "AND Gate" is used in the sense that R will occur as soon as S_1 and … or S_n occurs. Note that the kind number in the operator data is set to zero.

p. Type 11: m-out-of-n gate

(1) GO symbol:



(2) Operator data: 11, m, n, S_1, …, S_n, R

n: number of inputs, $2 \le n \le 10$

m: gate parameters, $1 \le m \le n$

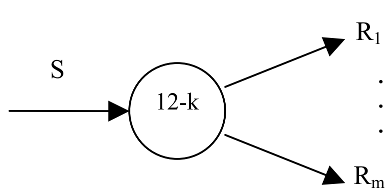
(3) Kind data: none

(4) Operation: Let V_1, V_2, …, V_n be the ordered set of values of VS_1, VS_2, …, VS_n (from smallest to largest). Then: $VR = V_m$

(5) Comments: Note that the kind number in the operator data is replaced with the gate parameter. If m = 1, this type is equivalent to a type 2; and if m = n, it is equivalent to a type 10.

q. Type 12: Path splitter

(1) GO symbol:



(2) Operator data: 12, K, S, m, R_1, …, R_m
m: number of outputs, $1 \le m \le 10$

(3) Kind data: K, 12, m, P_1, …, P_m
P_i: probability that i^{th} path is "selected"

$$\sum_i^m P_i = 1$$

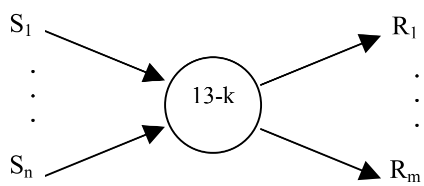
(4) Operation: m + 1 terms are produced. The first m of these are defined by:
VR_i = VS and VR_j = never for all $j \neq i$, with probability P_i, i = 1, …, m
And the m + 1^{st} is defined by:
VR_j = never for all j, with probability

$$1 - \sum_i^m P_i$$

(5) Comment: The m+1^{st} term does not occur if $\sum P_i = 1$.

r. Type 13: General purpose multiple input, multiple output operator

(1) GO symbol:



(2) Operator data: 13, K, n, S_1, …, S_n, m, R_1, …, R_m
n: number of inputs, $0 \le n \le 10$
m: number of outputs, $1 \le m \le 10$

(3) Kind data: K, 13, n, m, N

VS_{11} …$VS_{n1}m_1$
VR_1 … VR_mP_{11}
.
.
.
VR_1 … VR_m P_{m11}
VS_{1N} … $VS_{nN}m_N$
VR_1 … VR_m P_{1N}
.
.
.
VR_1 ... VR_m P_{mnN}

where:

n:	number of inputs, $0 \le n \le 10$
m:	number of outputs, $1 \le m \le 10$
N:	number of output time sets, $N \ge 1$, (if n = 0, N = 1)
Mi:	number of output terms for the i^{th} output time set
VS_{1i}, … VS_{ni}:	the ith input value comparison set (missing if n = 0)
P_{ij}:	probability of the i^{th} output term in the j^{th} output value set

$$\sum_{i=1}^{M_i} P_{ij} = 1,\ j = 1, \ldots, N$$

(4) Operation: If $n \neq 0$, the actual input values are compared with the N input value comparison sets. If a match is found, the corresponding joint output distribution is produced. If no match is found, all output values are set to never (with probability 1).
If $n = 1$, the signal joint output distribution is produced.

(5) Comments:

(a) The maximum amount of kind data is limited to 100 data items.

(b) For legibility the kind data should probably be laid out on several cards in the form indicated in c above rather than simply strung out item-by-item.

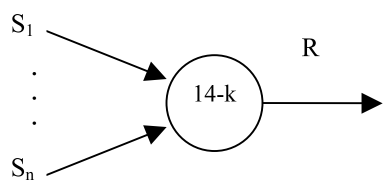
(c) In principle, any of the other GO types could be replaced by properly defined type 13's. However, the amount of kind data required for a complete definition is prohibitive in most cases. Consequently judicious use of the type 13 is indicated.

(d) Setting n (# of inputs) to zero gives us a signal generator which can produce several dependent signals (i.e., type 4) (as contrasted to several independent signals which would be produced by several type 5's).

(e) A type 13 can be easily used as a non-stochastic function device in which a single (multiple) output is defined as a function of a single (multiple) input.

s. Type 14: Linear combination generator

(1) GO symbol:



(2) Operator data: 14, K, n, $S_1, \ldots, S_n$, R
n: number of inputs, $2 \leq n \leq 10$

(3) Kind data: K, 14, n, $a_1, \ldots, a_n, a_0$
a_i: any real number

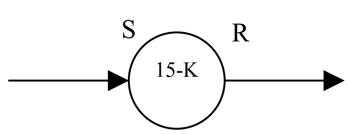
(4) Operation: Let A be the value of $a_0 + a_1 \times VS_1 + \ldots + a_n \times VS_n$ rounded to the nearest integer. Then:

VR = max 0, min A, never, if all $VS_i <$ never
VR = never, if any VS_i = never

(5) Comment: When using this type, signal values will usually be interpreted as amounts of some quantity rather than times.

t. Type 15: Time/probability gate-generator

(1) GO symbol:



(2) Operator data: 15, K, S, R

(3) Kind data: K, 15, $V_1, V_2, V_3, V_4, P_1, P_2$
V_1: output value if input is in gate (set to -1 if output value is to equal input value)
V_2: output value if input is not in gate
V_3, V_4: value gate values, $0 \leq V_3 \leq V_4 \leq$ never

P_1, P_2: probability gate values, $0 \le P_1 \le P_2 \le 1$

(4) Operation: Let $V = V1$, if $V1 \ge 0$

$= VS$, if $V1 = -1$

and PS = probability association with the input term

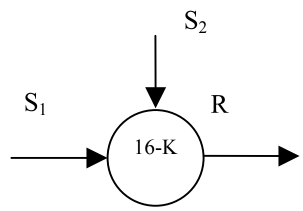
Then $VR = V$, if $V_3 \le VS \le V_4$ and $P_1 \le PS \le P_2$

$= V_2$, otherwise

(5) Comment: none

u. Type 16: Actuated normally open contact

(1) GO symbol:



(2) Operator data: 16, K, S_1, S_2, R

(3) Kind data: K, 16, P_1, P_2, P_3

P_1: contact operates normally

P_2: contact fails opened

P_3: contact fails closed

(4) Operation: $VR = 0$, if contact fails opened

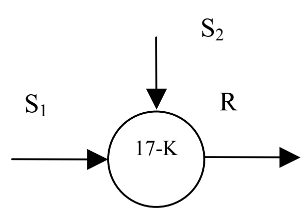
$= VS_1$, if contact fails closed

$= \min \{VS_1, VS_2\}$, if contact operates normally

(5) Comment: none

v. Type 17: Actuated normally closed contact

(1) GO symbol:



(2) Operator data: 17, K, S_1, S_2, R

(3) Kind data: K, 17, P_1, P_2, P_3

P_1: contact operates normally

P_2: contact fails closed

P_3: contact fails opened

(4) Operation: $VR = \infty$, if (a) contact normal and $VS_2 \ge VS_1$

(b) contact fails open and $VS_1 > 0$

(c) $VS_1 = 0$

$= 0$, if contact fails closed and $VS_1 > 0$

$= VS_2$, if $VS_1 = 0$ and $VS_2 < VS_1$ and contact normal

(5) Comment: Note the arbitrary convention that if $VS_1 = VS_2$, $VR = \infty$

w. Additional GO information. An understanding of the GO operator types and their algorithms is essential to modeling system availability or reliability appropriately. However, it is not necessary to be thoroughly familiar with all of the operator types since some are used more frequently than others. There is a natural hierarchy of the operator types, based on ease of use and utility. Experience has shown that practically all modeling situations can be handled with the first two groups of operators. But special situations and the sophistication of the modeling may result in the use of operators from the third group (least used types). The hierarchy is:

(1) Most commonly used types: 1, 2, 5, 6, 10

(2) Often used types: 3, 7, 9, 11, 15

(3) Least used types: 4, 8, 12, 13, 14, 16, 17

4-6. GO model development

As previously discussed, the node is the basic building block to any GO model. The node can represent either the physical equipment of the system or the logical equipment that ties the system together. Unfortunately, there is no way to model control loops in which feedback signals propagate from downstream components to upstream components. If the items in the control loop affect reliability, the influence of those items must be reduced to a series operator. GO also requires that all nodes be independent, if two or more components are not independent, their dependency can be modeled as a logical combination of independent nodes. In the end, the GO model will resemble a tree of nodes. Signals flow down through the tree until they reach the bottom or output nodes, which have no nodes connected to their outputs.

a. Step 1: One line drawing creation/analysis. The first step to performing an analysis with GO is to examine the one line drawing that represents the system. Often, the one line drawing must be developed by the analyst. The one line drawing provides the analyst the path that must be modeled by GO to accurately represent the physical and logical equipment of the system. Figure 4-1 represents a one line drawing of the IEEE Gold Book Standard Network System. This system is supplied by two independent 15kV primary distribution feeders. There are four diesel engine generators at the facility where two of four generators are required to meet the network load demands at all times. The reliability indices of the load points in figure 4-1 (i.e. OUTPUTS A, B1, B2, C, D, E1, E2) will be evaluated by the Boolean Algebra reliability analytical methodology. The following reliability indices will be evaluated:

(1) Frequency of load point interruptions (interruptions per year).

(2) Annual duration of load point interruptions (hours per year).

(3) Average duration of load point interruptions (hours per interruption).

(4) Availability level of power supply to the load point.

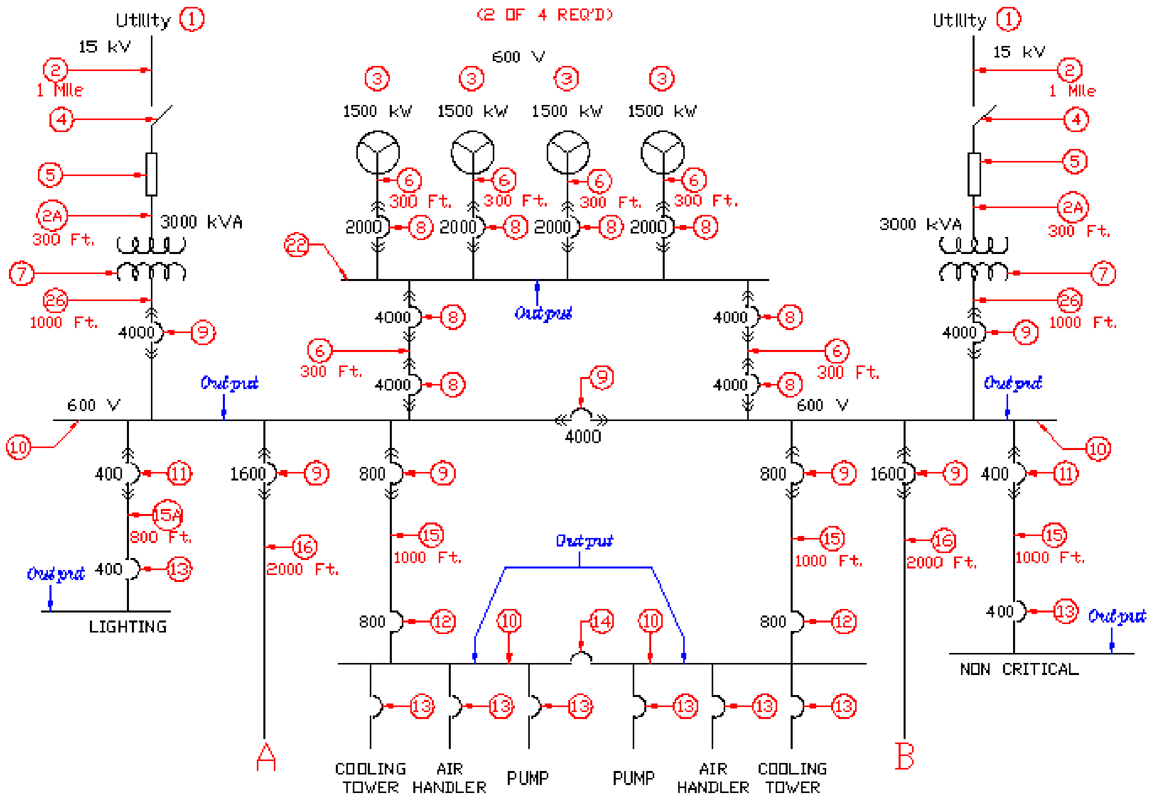


Figure 4-1. Single line diagram of IEEE Gold Book Standard Network.

b. Step 2: Parts identification. The next step is to develop a parts list of all the components found on the one line diagram. The analyst must identify the components compromising the system and identify them by functional categories. This way the reader can review the information in a logical process. Included in the parts list is the following information:

- Designation of the part identifying any alphanumeric reference on the one line diagram. This is the tie from the one line to the parts list.
- Description of the part including any qualifying information to ratings, size, normal operating position, etc.
- Identification of the Kind number that is a unique tracking identifier used in the GO model development.
- Reliability information including statistical numeric to develop the Inherent Availability information for the model.
- Data source category to track the different locations where the information originated.

(1) It is during this step that the analyst must determine what assumptions will be needed to complete the reliability analysis of the IEEE Gold Book Standard Network. The assumptions will allow the results that are obtained to be meaningfully compared with results obtained by using methodologies other than GO. The following assumptions were identified for any reliability methodology applied to the IEEE Gold Book Standard Network:

- Actual cable lengths are indicated on the drawings (see figure 4-1), modify failure rates accordingly. (For example, Cable Failure Rate per rated length X% of Actual Cable Length indicated on the drawing.)
- M denotes manual operation and is allocated 15 minutes for activation.
- 2 out of 4 generators are required.
- The UPS are redundant.
- The PDU transformers are redundant.
- Terminations, while normal for all systems, are omitted from the drawings. For this analysis terminations or splices are not included in the reliability calculations.
- Circuit breaker failure modes are assumed to be 50% open and 50% shorted.
- Constant failure rate is assumed.

(2) Table 4-2 identifies the pertinent statistical data used in the GO model to analyze the IEEE Gold Book Standard Network. The data is derived from the PREP database, which was discussed in paragraph 4-5b of this report, and supplemented by the IEEE Gold Book.

Table 4-2. Equipment availability data for Gold Book Standard Network configuration

Ref. #	Item Description	PREP Item #	Inherent Availability	MTTR (Hours)	Failure Rate (Failure/Year)	Calculated Availability
1	Single Circuit Utility Supply, 1.78 failures/unit years, A = 0.999705, Gold Book p. 107	NA	0.999705	1.32	1.956	?
2	Cable Arial, ≤ 15kV, per mile	32	0.99999022	1.82	0.047170	?
2A	Cable Arial, ≤ 15kV - 300 feet	32		1.82	0.002680	0.999999443
3	Diesel Engine Generator, Packaged, Stand-by, 1500kW	98	0.99974231	18.28	0.123500	?
4	Manual Disconnect Switch	187	0.9999998	1	0.001740	?
5	Fuse, 15kV	117	0.99995363	4	0.101540	?
6	Cable Below Ground in conduit, ≤ 600V, per 1000 ft	47	0.99999743	11.22	0.002010	?
6A	Cable Below Ground in conduit, ≤ 600V - 300 feet			11.22	0.000603	0.999999228
7	Transformer, Liquid, Non Forced Air, 3000kVA	208	0.99999937	5	0.001110	?
8	Ckt. Breaker, 600V, Drawout, Normally Open, > 600 Amp	68	0.99999874	2	0.005530	?
8A	Ckt. Breaker, 600V, Drawout, Normally Open, > 600 Amp	68		2	0.002765	0.999999369
9	Ckt. Breaker, 600V, Drawout, Normally Closed, >600 Amp	69	0.99999989	0.5	0.001850	?
9A	Ckt. Breaker, 600V, Drawout, Normally Closed, >600 Amp	69		0.5	0.000925	0.999999947
10	Switchgear, Bare Bus, 600V	191	0.9999921	7.29	0.009490	?
11	Ckt. Breaker, 600V Drawout, Normally Closed, < 600 Amp	67	0.99999986	6	0.000210	?
11A	Ckt. Breaker, 600V Drawout, Normally Closed, < 600 Amp	67		6	0.000105	0.999999928
12	Ckt. Breaker, 600V, Normally Closed, > 600 Amp, Gold Book p. 40	63	0.99998948	9.6	0.009600	?
12A	Ckt. Breaker, 600V, Normally Closed, > 600 Amp, Gold Book p. 40	63		9.6	0.004800	0.999994740
13	Ckt. Breaker, 3 Phase Fixed, Normally Closed, ≤ 600 Amp	61	0.99999656	5.8	0.005200	?
13A	Ckt. Breaker, 3 Phase Fixed, Normally Closed, ≤ 600 Amp, Gold Book p. 40	61		5.8	0.002600	0.999998279
14	Ckt. Breaker, 3 Phase Fixed, Normally Open, > 600 Amp	62	0.99998532	37.5	0.003430	?
14A	Ckt. Breaker, 3 Phase Fixed, Normally Open, > 600 Amp	62		37.5	0.001715	0.999992658
15	Cable, Above Ground, No Conduit, ≤ 600V, per 1000 ft.	20	0.99999997	2.5	0.000120	?
15A	Cable, Above Ground, No Conduit, ≤ 600V, per 1000 ft.	20		2.5	0.000096	0.999999973
16	Cable, Above Ground, Trays, ≤ 600V, per 1000 ft., Gold Book p.105		0.99999831	10.5	0.001410	?
	Cable, Above Ground, Trays, ≤ 600V, per 1000 ft., Gold Book p.105			10.5	0.002820	0.999996620
22	Switchgear, Insulated Bus, ≤ 600V		0.99999953	2.4	0.001700	0.999999534
26	Bus Duct, Gold Book p. 206, per Circuit foot		0.99999982	12.9	0.000125	0.999815959

c. Step 3: Logic model development. The third step in the development of the GO model is to produce a logical representation of the one line diagram. This model will provide the functional relationship to the system. Figure 4-2 shows the resulting Boolean Algebra model. In this process you will use the Kind number identified in the parts list from table 4-2 as a unique identifier. This unique identifier is then combined with the logical operators outlined in paragraphs 4-5 through 4-5u to create a functional model. Functional paths are developed to show the operational characteristics of the system and these paths are identified with signal inputs and outputs.

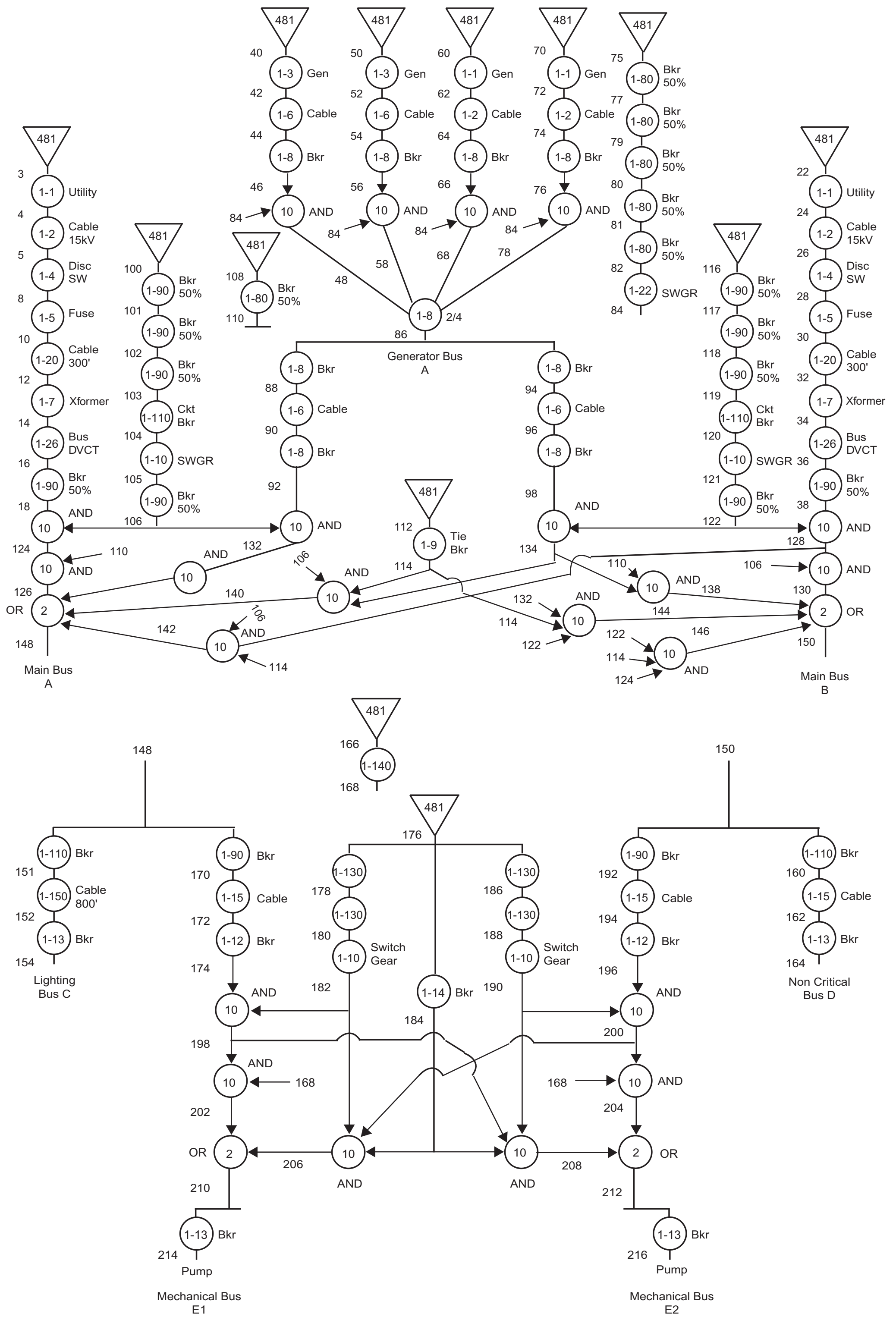


Figure 4-2. Boolean algebra diagram of the IEEE Gold Book Standard Network.

(1) Figure 4-2 illustrates the complete representation of the IEEE Gold Book Standard Network, but the following figures and description outline how the model was built. Figure 4-3 shows how the utility leg that appears on the left of figure 4-2 was created. This utility leg starts with a type 5 signal generator that is given a kind number of 481. The output signal of this perfect start is 2 which then becomes the input to the utility, modeled as a type 1 operator with a kind number of 1-1. The utility that emits a signal of 4 to the 15kV cable represented by another type 1 operator with kind number of 1-2. A 6 signal then connects the cable with a manual disconnect switch (type 1 operator, kind number 1-4) that outputs an 8 signal to a fuse (type 1 operator, kind number 1-5). Next, a 10 signal connects the fuse with another cable (type 1 operator, kind number 1-20) and emits a signal of 12 into a transformer (type 1 operator, kind number 1-7). The signal, 14, from the transformer then enters a bus duct (type 1 operator, kind number 1-26) continues as signal 16 into a circuit breaker (type 1 operator, kind number 1-9) before combination via an AND gate (type 10 operator, kind number 10) as signal 18. This path continues combining with other components of the system as it proceeds toward main bus A and beyond.

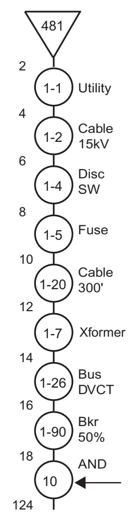


Figure 4-3. Utility 1 path to main bus A.

(2) Figure 4-4 describes the signal path at the top-center of figure 4-2 as the generators are combined via a 2 of 4 requirement. Four similar signal paths are started for each generator with perfect starts, type 5 operator with kind number of 481. The perfect starts output signals (40, 50, 60, and 70 respectively for each path) to the generators (type 1 operators, kind numbers 1-3). The generators then outputs signals (42, 52, 62, and 72 respectively) to cables (type 1 operators, kind numbers 1-6), the signal path (now as 44, 54, 64, and 74 respectively) continues into breakers (type 1 operators, kind number 1-8). At this point the signals, 46, 56, 66, and 76 are each combined via an AND gate with the switchgear and circuit breaker path of signal 84. Signal 84 is necessary to represent the capability of the breakers contributing to a bus shut down by passing the fault back to the generators. Fifty percent contribution to this failure mode of failure to open is estimated for this analysis. The AND gate output signals of 48, 58, 68, and 78 are then combined via a M out of N gate (type 11 operator, kind number 11) with the 2 of 4 requirement. The single output signal from the M out of N gate then enters the generator bus tiebreaker as signal 86, which splits to go down two different breaker paths that will create the tiebreaker effect as either side can be used to provide a signal to the components downstream. The steps identified to create the figures 4-3 and 4-4 can be used to model the remaining components and operators of the system.

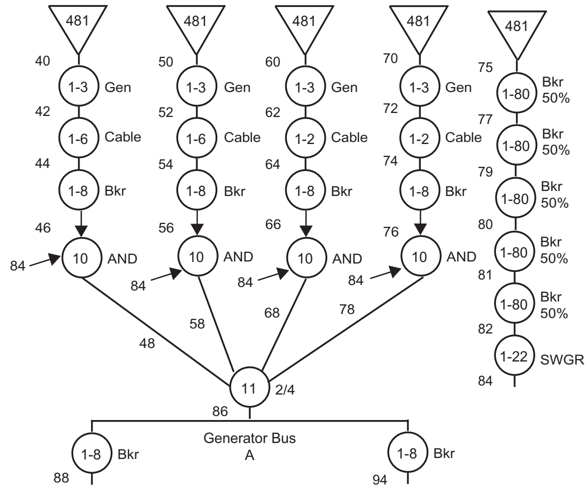


Figure 4-4. Paths to generator bus.

d. Step 4: Input file creation. The next step is to assemble the files that will be used as an input to the GO model. The GO software tool requires three main files that must be specifically configured to run in the program. These files are the model file, GO1.in file, the parts list file, GO2.in file, and the results file, GO3.in file. These files are normally created using the Notepad accessory on most computers.

(1) Utilizing the model developed in the previous step, a GO1 file is assembled based on the illustrated signal path of figure 4-2 with the identified Kind number representing the component. Operators are also inserted in the model path to represent the logical flow of operation. Table 4-3 contains the GO1 file that was developed to represent the IEEE Gold Book Standard Network. This file contains the components and operators representing the functional operation of the system. The last line of this file represents the desired output signals that an availability metric will be calculated (operator type of 0).

Table 4-3. GO1 model file

```
PREP Recommended Power Plant Model
 $param infin=1$
5 481 2                     $
1 1 2 4                     $
1 2 4 6                     $
1 4 6 8                     $
1 5 8 10                    $
1 20 10 12                  $
1 7 12 14                   $
1 26 14 16                  $
1 90 16 18                  $
5 481 22                    $
1 1 22 24                   $
1 2 24 26                   $
1 4 26 28                   $
1 5 28 30                   $
1 20 30 32                  $
1 7 32 34                   $
1 26 34 36                  $
1 90 36 38                  $
5 481 40                    $
1 3 40 42                   $
1 6 42 44                   $
1 8 44 46                   $
5 481 50                    $
1 3 50 52                   $
1 6 52 54                   $
1 8 54 56                   $
5 481 60                    $
1 3 60 62                   $
1 6 62 64                   $
1 8 64 66                   $
5 481 70                    $
1 3 70 72                   $
1 6 72 74                   $
1 8 74 76                   $
5 481 75                    $
1 80 75 77                  $
1 80 77 79                  $
1 80 79 80                  $
1 80 80 81                  $
1 80 81 82                  $
1 22 82 84                  $
10 0 2 46 84 48             $
10 0 2 56 84 58             $
10 0 2 66 84 68             $
10 0 2 76 84 78             $
11 2 4 48 58 68 78 86       $
1 8 86 88                   $
1 6 88 90                   $
1 8 90 92                   $
1 8 86 94                   $
1 6 94 96                   $
1 8 96 98                   $
```

Table 4-3. GO1 model file (continued)

```
5 481 100                    $
1 90 100 101                 $
1 90 101 102                 $
1 90 102 103                 $
1 110 103 104                $
1 10 104 105                 $
1 90 105 106                 $
5 481 108                    $
1 80 108 110                 $
5 481 112                    $
1 9 112 114                  $
5 481 116                    $
1 90 116 117                 $
1 90 117 118                 $
1 90 118 119                 $
1 110 119 120                $
1 10 120 121                 $
1 90 121 122                 $
10 0 2 18 106 124            $
10 0 2 124 110 126           $
10 0 2 38 122 128            $
10 0 2 110 128 130           $
10 0 2 92 106 132            $
10 0 2 98 122 134            $
10 0 2 110 132 136           $
10 0 2 134 110 138           $
10 0 3 134 106 114 140       $
10 0 3 128 106 114 142       $
10 0 3 132 114 122 144       $
10 0 3 114 122 124 146       $
2 0 4 126 136 140 142 148    $
2 0 4 130 138 144 146 150    $
1 110 148 151                $
1 150 151 152                $
1 13 152 154                 $
1 110 150 160                $
1 15 160 162                 $
1 13 162 164                 $
5 481 166                    $
1 140 166 168                $
1 90 148 170                 $
1 15 170 172                 $
1 12 172 174                 $
5 481 176                    $
1 130 176 178                $
1 130 178 180                $
1 10 180 182                 $
1 14 176 184                 $
1 130 176 186                $
1 130 186 188                $
1 10 188 190                 $
1 90 150 192                 $
1 15 192 194                 $
1 12 194 196                 $
10 0 2 174 182 198           $
10 0 2 190 196 200           $
10 0 2 198 168 202           $
10 0 2 168 200 204           $
10 0 3 182 184 200 206       $
10 0 3 184 190 198 208       $
2 0 2 202 206 210            $
2 0 2 204 208 212            $
1 13 210 214                 $
1 13 212 216                 $
0 18 38 126 130 148 150 128 124 210 212
86 154 164   $
```

(2) The GO2 file contains the reliability information that the GO1 file acts upon. For each Kind number defined in the GO1 file there is an availability metric representing the component's individual availability. The GO1 model assembles the individual availability information for each component with the logical operators to determine the overall system availability. Table 4-4 identifies the GO2 parts list file used to identify the individual availability data.

Table 4-4. GO2 parts list file

```
 1   1      1 .999705     .000295     $Single Circuit Utility Supply, 1.78 failures/u
 2   2      1 .999990218  .000009782  $Cable Aerial, <= 15kV, per mile
 3   20     1 .999999443  .000000557  $Cable Aerial, <= 15kV, per mile, 300 ft.
 4   3      1 .999742312  .000257688  $Diesel Engine Generator, Packaged,Stand-by, 15
 5   4      1 .999999801  .000000199  $Manual Disconnect Switch
 6   5      1 .999953634  .000046366  $Fuse,  15kV
 7   6      1 .999997428  .000002572  $Cable Below Ground in conduit, <=600V, per 100
 8   60     1 .999999228  .000000772  $Cable Below Ground in conduit, <=600V, per 100
 9   7      1 .999999367  .000000633  $Transformer, Liquid, Non Forced Air, 3000kVA
 10  8      1 .999998738  .000001262  $Ckt. Breaker, 600v, Drawout, Normally Open, >
 11  80     1 .999999369  .000000631  $Ckt. Breaker, 600v, Drawout, Normally Open, >
 12  9      1 .999999894  .000000106  $Ckt. Breaker, 600V, Drawout, Normally Closed,>
 13  90     1 .999999947  .000000053  $Ckt. Breaker, 600V, Drawout, Normally Closed,>
 14  10     1 .999992098  .000007902  $Switchgear, Bare Buss, 600V
 15  11     1 .999999858  .000000142  $Ckt. Breaker, 600v Drawout, Normally Closed, <
 16  110    1 .999999928  .000000072  $Ckt. Breaker, 600v Drawout, Normally Closed, <
 17  12     1 .999989479  .000010521  $Ckt. Breaker, 600V, Normally Closed, >600 Amp,
 18  120    1 .999994740  .000005260  $Ckt. Breaker, 600V, Normally Closed, >600 Amp,
 19  13     1 .999996557  .000003443  $Ckt. Breaker, 3 Phase Fixed, Normally Closed,
 20  130    1 .999998278  .000001722  $Ckt. Breaker, 3 Phase Fixed, Normally Closed,
 21  14     1 .99998532   .00001468   $Ckt. Breaker, 3 Phase Fixed, Normally Open, >6
 22  140    1 .999992658  .000007342  $Ckt. Breaker, 3 Phase Fixed, Normally Open, >6
 23  15     1 .999999966  .000000034  $Cable, Above Ground, No Conduit, <= 600V, per
 24  150    1 .999999973  .000000027  $Cable, Above Ground, No Conduit, <= 600V, per
 25  16     1 .99999831   .00000169   $Cable, Above Ground, Trays, <= 600V, per 1000
 26  160    1 .999996620  .000003380  $Cable, Above Ground, Trays, <= 600V, per 1000
 27  22     1 .999999534  .000000466  $Switchgear, Insulated Buss, <=600V
 28  26     1 .999815958  .000184042  $Bus Duct, Gold Book p.206, per Circuit foot, 1
 29  480    1 .995000000  .005000000  $Manual Operator
 30  481    5 1 0 1.0                 $Perfect Start
```

(3) The GO3 file identifies the parameters that govern the calculation of the availability results for the desired output signals. This file usually contains no data since its sole purpose is to define the results obtained from the executable files. Normally this file always contains the same information as illustrated in table 4-5.

Table 4-5. GO3 results file

```
GO-3 DATA
 $PARAM PMIN=1.e-13 $
```

e. Step 5: Performing analysis. There are three executable files that comprise the GO software, GO1.exe, GO2.exe, and GO3.exe. The first two executable files are used in conjunction with the GO1 and GO2 input files, respectively. These executable files will read the input files and develop the necessary output files leading up to the system availability analysis. The final executable file provides the system availability results when all files are executed successfully. This file contains the signal output(s) with their associated availability metric(s). Since GO is MS-DOS based program the executables and input files must be located within the same folder and the folder must be accessible while in a MS-DOS mode. The three executable files all write output text files into the folder where the input and executable files lie (output files are named "t1," "t2," and "t3" to coincide with their respective executable file). The following example, see figure 4-5, will illustrate how an analysis is performed (in this case the executable and input files are located on the C:\ drive in a folder labeled "GO").

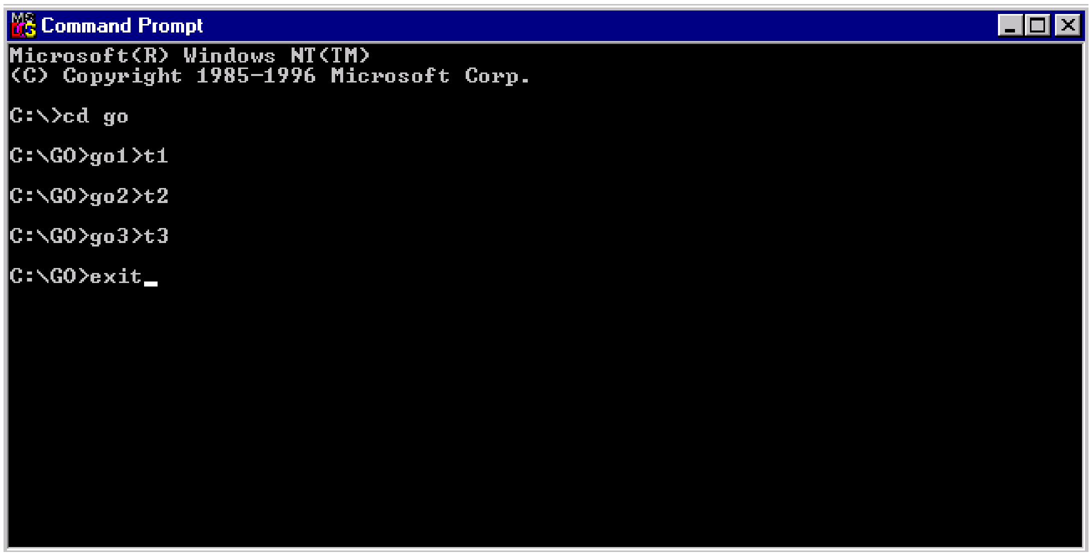


Figure 4-5. MS-DOS screen while performing analysis.

(1) Figure 4-5 was taken just prior to returning to the MS Windows environment from the MS-DOS window. As illustrated in the figure, the steps taken to run the availability analysis consist of:

- Opening folder where executable and input files exist (cd go).
- Run input model GO1 to output model t1 (go1>t1).
- Execute input model GO2 to output model t2 (go2>t2).
- Run results file, GO3, to output model t3 (go3>t3).
- Exit MS-DOS to return to MS Windows environment.

(2) The output files, t1 and t2, for the GO1 model file and GO2 parts list file contain all the information from the input files plus additional information that GO uses to analyze the system. The t1 file also includes information regarding each signal called out within the model, which is used to identify the logic of the model and determine if signals have an output, if not they are added to final signal list that was identified at end of GO1 model file. The t2 output file identifies the parts list from GO2 and identifies how many of each operator and kind type are called out in the GO1 model file.

(3) The t3 output file contains the results of the GO analysis with each output signal given an availability metric. This output file is shown in table 4-6.

Table 4-6. GO results output file

```
GO-3 DATA

OPERATOR FILE -

PREP Recommended Power Plant Model     KIND FILE ---- GO2 Input
----------------------------------------------------------------

TOTAL PROBABILITY =  .9999999999871
TOTAL ERROR =        .0000000000129

INDIVIDUAL SIGNAL PROBABILITY DISTRIBUTIONS

VAL.        148
----   -------------
  0   .9999912359880
  1   .0000087639992

VAL.        150              210              212
----   -------------    -------------    -------------
  0   .9999912359883   .9999886527944   .9999886527944
  1   .0000087639989   .0000113471928   .0000113471928

VAL.         86              154              164
----   -------------   --------------    -------------
  0   .9999963789276   .9999876940199   .9999876870202
  1   .0000036210596   .0000123059672   .0000123129669
```

f. Step 6: Troubleshooting. Unfortunately, GO models rarely yield the desired availability results on the first attempt to perform the analysis. Therefore, analysts will need to understand the potential pitfalls associated with the GO software and the input files required to perform the analysis. Following are many of the known pitfalls associated with a GO availability analysis.

(1) All executable files (GO1.exe, GO2.exe, and GO3.exe) and GO input files (GO1.in, GO2.in, and GO3.in) must be located in the same folder and be accessible from the MS-DOS environment on the computer.

(2) The contents of the GO1 and GO2 input files must be meticulously entered to ensure model is accurately portrayed. See table 4-7.

Table 4-7. Accurately entering GO1 and GO2 files

In the case of GO1 this means making sure that the operator types, kind numbers, and signals are entered correctly. • Special attention should be given to ensure that all signals that are output signals of one component are the input signal of the next component or operator in series unless it is to be included as a final signal. • Unless using a generator operator an input signal can not be referred to unless already defined as the output signal of another operator. • A check should be made to verify that signals are not re-used in the model also. • The order of information for this file is the operator type to the far left followed by a space, then the kind number, another space, the input signal, another space, the output signal, another space, and the description beginning with a $ sign. The previous description is for type 1 Operators only, refer to paragraph 4.1 for alternative operators. • The maximum number of final signals that can be analyzed are 16. • The signal number can not exceed 8999.
GO2 requires a standard format with the following guidelines. • The kind numbers line up on the far left with a certain number of spaces between the kind number and operator type (6 for single digit kind numbers, 5 for double digits, etc.), then one space to 9 decimal availability values for kind numbers followed by two spaces and 9 decimal unavailability values, and finally two spaces until the description portion that begins with a $ sign. The previous description is for type 1 Operators only, refer to paragraph 4.1 for alternative operators. • The availability and unavailability metric can not start with a 0 prior to the decimal. • These availability and unavailability metrics must add up to 1.000000000.

(3) If a problem arises during the analysis usually an error message will occur following the step that is being completed. Table 4-8 lists these error messages.

Table 4-8. Analysis error messages

Run-time error F6501: READ(OPGO1.XXX)," followed by "- end of file encountered," on the next line for errors within the GO1 input file as the GO1 executable is being run to t1 file. Run-time error F6600: WRITE(internal)," followed by "- internal file overflow," on the next line for errors within the GO2 input file.

(4) The best means of identifying a problem is to examine the output files, t1, t2, and t3. First examine the output file corresponding to the last step that was being performed when the error appeared and work back through other files (i.e. if GO2.exe encountered an error examine t2 then t1 files). If the corresponding executable ran successfully there will be a note at the bottom of the output file stating "GO# FINISHED" (the # corresponds to the output file being examined). If an error occurred it will state "SUICIDE BECAUSE OF ERRORS," followed by "FATAL ERROR: …..SUICIDE…..," two lines later and the error messages shown in table 4-9 will commonly be found within the output files.

Table 4-9. Error messages

"**** SIGNAL X REUSED," followed by "------ERROR------" in the next line of the t1 file, this message reflects the identification of a signal number as an output signal for more than one area of the model.
"**** INPUT SIGNAL X HAS NOT BEEN ENTERED," followed by "------ERROR------" in the next line, this error message appears in the t1 file. This message identifies that an input signal was called out in the model even though it has not been previously modeled as an output signal for another component within the model.
"**** THERE ARE TOO MANY FINAL SIGNALS," appears in the t1 file as signals not used as inputs to other operators are added to final signal list. This message is yielded when more than 16 final signals are identified (either within the last line of GO1 input file or by GO software when GO1 file is executed and all end signals become final signals).
"**** PROBABILITY SUM IS X," within the t2 file signifies that the reliability/unreliability or availability/unavailability numeric combinations specified for an operator does not add up to 1.